

In the Claims:

Please amend claims 1, 15, 18, 27 and 30, and please cancel claims 11, 17 and 29, as indicated below.

1. (Currently amended) A thread-safe scheduler system, comprising:

at least one processor; and

memory coupled to the at least one processor, wherein the memory is configured to store program instructions executable by the at least one processor to implement:

a primary scheduler which is executable to schedule requests for networked data resources; and

a secondary scheduler, wherein the secondary scheduler is executable to receive a plurality of requests from a multi-threaded application in a thread-safe manner and send the requests received from the multi-threaded application to the primary scheduler in a thread-safe manner;

wherein the requests comprise callback functions, and wherein the callback functions are executable to send responses to the requests to the multi-threaded application.

2. (Original) The system of claim 1, wherein the primary scheduler is single-threaded.

3. (Original) The system of claim 1, wherein the secondary scheduler is multi-threaded.

4. (Original) The system of claim 1, wherein the secondary scheduler is executable to receive the plurality of requests from the multi-threaded application through a lock in a thread-safe manner.

5. (Original) The system of claim 1, wherein the primary scheduler is executable to receive the plurality of requests from the secondary scheduler through a lock in a thread-safe manner.

6. (Previously presented) The system of claim 1, wherein the resources comprise a management information server, wherein the requests received from the multi-threaded application comprise management requests, and wherein the multi-threaded application comprises a multi-threaded manager application.

7. (Previously presented) The system of claim 1, wherein the program instructions are executable by the at least one processor to further implement:

a management information server coupled to the primary scheduler through a management interface, wherein the primary scheduler is operable to send the requests from the secondary scheduler to one or more managed objects through the management information server.

8. (Original) The system of claim 7, wherein the managed objects comprise one or more objects corresponding to a telephone network.

9. (Original) The system of claim 7, wherein the managed objects comprise an object corresponding to a telecommunications device.

10. (Original) The system of claim 7, wherein the management interface comprises Portable Management Interface (PMI), wherein PMI is single-threaded, and wherein PMI comprises a plurality of functions which are operable to carry out the

requests.

11. (Canceled)

12. (Original) The system of claim 1, wherein the primary scheduler comprises a primary queue which is operable to hold pending requests and responses to the requests.

13. (Original) The system of claim 1, wherein the secondary scheduler comprises a secondary queue which is operable to hold pending requests.

14. (Previously presented) The system of claim 1, wherein the program instructions are executable by the at least one processor to further implement:

a communication pipe between the primary scheduler and secondary scheduler,
wherein the secondary scheduler uses the communication pipe to wake the
primary scheduler prior to sending one of the requests to the primary
scheduler.

15. (Currently amended) A thread-safe method for using a management interface for management of a plurality of managed objects on a network, the method comprising:

receiving a plurality of management requests from a multi-threaded manager
application into a secondary scheduler in a thread-safe manner;

scheduling the plurality of management requests in a secondary queue in the
secondary scheduler after receiving the management requests from the
manager application;

sending the management requests from the secondary scheduler to a primary
scheduler in a thread-safe manner;

scheduling the management requests in a primary queue in the primary scheduler,
wherein each of the management requests comprises a corresponding
callback function; and

executing the management requests on the managed objects after scheduling the
management requests in the primary queue.

16. (Original) The method of claim 15, wherein executing the management
requests on the managed objects further comprises sending the management requests to a
management information server coupled to the managed objects.

17. (Canceled)

18. (Currently amended) The method of claim ~~17~~ 15, further comprising:

receiving a response to one of the management requests from one of the managed
objects after executing that management requests on one of the managed
objects; and

executing the corresponding callback function for that management request to
send the response to the multi-threaded manager application.

19. (Original) The method of claim 18, further comprising:

enqueueing the response in the primary queue after receiving the response from
one of the managed objects;

finding the callback function corresponding to the response after enqueueing the
response; and

dequeuing the response from the primary queue before executing the corresponding callback function to send the response to the multi-threaded manager application.

20. (Original) The method of claim 15, wherein the management interface comprises Portable Management Interface (PMI), wherein PMI is single-threaded, and wherein PMI comprises a plurality of functions which are operable to execute the management requests.

21. (Original) The method of claim 15, wherein the managed objects comprise one or more objects corresponding to a telephone network.

22. (Original) The method of claim 15, wherein the managed objects comprise an object corresponding to a telecommunications device.

23. (Original) The method of claim 15, wherein the receiving the plurality of management requests from the multi-threaded manager application into the secondary scheduler in the thread-safe manner comprises receiving the plurality of management requests through a thread-safe lock.

24. (Original) The method of claim 15, wherein sending the management requests to the primary scheduler in the thread-safe manner comprises dispatching the management requests through a thread-safe lock.

25. (Original) The method of claim 15, wherein the primary scheduler is executed in a single thread associated with the management interface, and wherein the secondary scheduler is executed in at least one different thread.

26. (Original) The method of claim 15, wherein the secondary scheduler is multi-threaded.

27. (Currently amended) A tangible, computer-readable storage medium comprising program instructions, wherein the program instructions are computer-executable to implement:

receiving a plurality of management requests from a multi-threaded manager application into a secondary scheduler in a thread-safe manner;

scheduling the plurality of management requests in a secondary queue in the secondary scheduler after receiving the management requests from the manager application;

sending the management requests from the secondary scheduler to a primary scheduler in a thread-safe manner;

scheduling the management requests in a primary queue in the primary scheduler, wherein each of the management requests comprises a corresponding callback function; and

executing the management requests on the managed objects after scheduling the management requests in the primary queue.

28. (Previously presented) The tangible, computer-readable medium of claim 27, wherein executing the management requests on the managed objects further comprises sending the management requests to a management information server coupled to the managed objects.

29. (Canceled)

30. (Currently amended) The tangible, computer-readable medium of claim ~~29~~ 27, wherein the program instructions are further executable to implement:

receiving a response to one of the management requests from one of the managed objects after executing that management request on one of the managed objects; and

executing the corresponding callback function for that management request to send the response to the multi-threaded manager application.

31. (Previously presented) The tangible, computer-readable medium of claim 30, wherein the program instructions are further executable to implement:

enqueueing the response in the primary queue after receiving the response from one of the managed objects;

finding the callback function corresponding to the response after enqueueing the response; and

dequeueing the response from the primary queue before executing the corresponding callback function to send the response to the multi-threaded manager application.

32. (Previously presented) The tangible, computer-readable medium of claim 27, wherein the management interface comprises Portable Management Interface (PMI), wherein PMI is single-threaded, and wherein PMI comprises a plurality of functions which are operable to execute the management requests.

33. (Previously presented) The tangible, computer-readable medium of claim 27, wherein the managed objects comprise one or more objects corresponding to a telephone network.

34. (Previously presented) The tangible, computer-readable medium of claim 27, wherein the managed objects comprise an object corresponding to a telecommunications

device.

35. (Previously presented) The tangible, computer-readable medium of claim 27, wherein the receiving the plurality of management requests from the multi-threaded manager application into the secondary scheduler in the thread-safe manner comprises receiving the plurality of management requests through a thread-safe lock.

36. (Previously presented) The tangible, computer-readable medium of claim 27, wherein sending the management requests to the primary scheduler in the thread-safe manner comprises dispatching the management requests through a thread-safe lock.

37. (Previously presented) The tangible, computer-readable medium of claim 27, wherein the primary scheduler is executed in a single thread associated with the management interface, and wherein the secondary scheduler is executed in at least one different thread.

38. (Previously presented) The tangible, computer-readable medium of claim 27, wherein the secondary scheduler is multi-threaded.